REMARKS

This is a reply to the Non-Final Office action of September 14, 2010. Claims 1-22 are pending. Applicants request reexamination and reconsideration of the application.

In sections 1-2, the Office action raises new grounds to reject claims 1 - 22 as obvious over U.S. Patent No. 6,785,769 B1 to Jacobs et al. (Jacobs) in view of U.S. Patent No. 6,366,947 B1 to Kavner (Kavner), and further in view of U.S. Published Patent Application No. 2002/0032701 A1 to Gao et al. (Gao).

However, Jacobs, Kavner, and Gao fail to describe claim 1:

A client-side caching system, comprising:

a client for issuing a request based on a user selection for a resource on a server; and

a server for receiving the request and sending a response including a cookie and a client-side script to the client, wherein the cookie value represents the last version of the resource, and wherein the client-side script appends the cookie value to the request for the resource and causes the client to automatically re-request the resource with the appended cookie value so that if the last version of the resource is in the client cache, the resource is retrieved from the client cache rather than from the server, and if not, the resource is retrieved from the server.

Jacobs' describes public caching of web pages and extracting a data identifier (e.g., URL, URI, address, file name, or policy) from a request to identify a suitable web page for each user. For example, if a user speaks English, the server provides a web page in English rather than Spanish. Or if the browser wants a web page without frames, a web page without frames is served, etc. (See Jacobs col. 1).

Further, Jacobs fails to describe at least the following limitations of claim 1:

1) A client-side cache system. Instead, Jacob describes a public web cache system 104 located between a client and server. It's a public cache accessed by multiple clients such as client 108a and 108b shown in Figure 1 (See Jacobs' col. 3, line 52 - col. 6, line 65).

2) A server sending a client-side script to a client.

3) A cookie value representing the last version of the resource such as a web page. Jacobs never suggests a cookie represent the last version, but merely information to select a suitable language, frame, or protocol for a client (See Jacobs' col. 1, lines 10-21).

4) A client-side script that appends the cookie value to the request for the resource and causes the client to automatically re-request the resource with the appended cookie value so that if the last version of the resource is in the client cache, the resource is retrieved from the client cache rather than from the server, and if not, the resource is retrieved from the server as recited in claim 1.

Kavner also fails to describe claim 1. Kavner displays a requested page stored in browser cache (even if out-of-date) and uses the If-Modified-Since header in the background to tell the server to send the page only if it has been modified since the time the page was cached (col. 10, line 46 – col. 11, line 7). Because a stale page will be immediately displayed, Kavner suggests using a beep to notify a user the page is stale until it's updated. Kavner must like beeps more than me.

Although the Office action looks to Gao to fill these gaps, Gao fails to describe the client-side script function required in claim 1.

1) Gao never suggests a client-side script causing a client to automatically re-request the resource with the appended cookie value so that if the last version of the resource is in the client cache, the resource is retrieved

from the client cache rather than from the server, and if not, the resource is retrieved from the server as recited in claim 1.

2) The Office action argued Gao teaches "a client-side script that automatically requests updated data" but claim 1 never recites this. Instead, claim 1 recites a client for issuing a request based on a user selection for a resource on a server and a client-side script that causes the client to automatically re-request the resource with the appended cookie value.

3) Gao never re-requests any resource. As shown in Figure 5, Gao makes a single request for a URL page (entry "1"), a single request for each of the images (entry "3"), and a single request of an "update page" (entry "5"). To assert Gao's client-side script re-requests any of these resources has no support.

4) To "request updated data" is not the same as to "re-request the resource ..." as recited in claim 1. *See, e.g., In re Suitco Surface, Inc.,* Slip Opinion 2009-1418 (Fed. Cir. 2010) ("The broadest-construction rubric ... does not give the PTO an unfettered license to interpret claims to embrace anything remotely related to the claimed invention. Rather, claims should always be read in light of the specification and teachings in the underlying patent.")

Because the client-side script function recited in claim 1 is absent in Jacobs, Kavner, and Gao, they cannot establish a prima facie case of obviousness. *See In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988) which held a reference did not render the claimed combination prima facie obvious because *the Patent Office ignored that a claim limitation was absent in the reference.*

The Office action ignores the limitations of claim 1 that are absent in Jacobs, Kavner, and Gao by simply deleting them. To show how extensively claim 1 was revised, we use strike-through for deleted limitations and underline for added limitations (See pages 2-3 of the Office action):

A client-side caching system, comprising:

a client for issuing a request based on a user selection for a resource on a server; and

a server for receiving the request and sending a response including a cookie ~~and a client-side script to the client~~, wherein the cookie value represents the last version of the resource, and wherein the cookie value is attached ~~client-side script appends the cookie value~~ to the request for the resource and ~~causes the client to automatically re-request~~ the client requests the resource with the appended cookie value so that if the last version of the resource is in the client cache, the resource is retrieved from the client cache rather than from the server, and if not, ~~the resource~~ is retrieved from the server.

Although a few deleted terms (e.g., client and script) come back on page 3, the other extensive revisions to claim 1 involve legal error. A judge cannot add or subtract words from the claims, i.e., rewrite the claims. *See, e.g., Callicrate v. Wadsworth Mfg., Inc.*, 427 F.3d 1361, 1369, 77 USPQ2d 1041 (Fed. Cir. 2005). *SmithKline Beecham Corp. v. Apotex Corp.*, 403 F.3d 1331, 1339-1340 74 USQP2d 1396 (Fed. Cir. 2005), *cert. denied*, 547 U.S. 1218 (2006). Similarly, an examiner can't determine patentability apart from examining the actual language of claim 1.

Jacobs' public cache serving multiple clients cannot be modified to be a client-side cache without completing defeating its purpose. It is hindsight to suggest a public cache system should be a client-side cache because it is more efficient. Moreover, even if Jacobs, Kavner, and Gao were combined, they do not teach the client-side function recited in claim 1. Thus, claim 1 would not have been obvious to the person of ordinary skill in the art. Thus, claim 1 is allowable over Jacobs, Kavner and Gao, and claims 2-4 and 8 are allowable due to their dependency on claim 1.

Contrary to the Office action on page 4, Jacobs fail to teach the client-side caching system of claim 1, wherein the response includes a non-displayed relatively small page and a client-side script is in the entity body of the response

as recited in claim 3. Jacobs describes a web cache rather than a client-side caching system. Jacobs col. 4, lines 25-36 fail to teach a non-displayed relatively small page and a client-side script in the entity body of the response.

Contrary to the Office action on page 4, Jacobs fails to teach the client-side caching system of claim 1, wherein the client-side script that appends the cookie value to the request is embedded in a displayed page as recited in claim 4. Instead, Jacobs col. 8, lines 10-16 mention a cookie appended to a data item identifier (e.g., URL).

Claim 5 is patentable over Jacobs, Kavner, and Gao for the reasons presented above with respect to claim 1.

Contrary to the Office action on page 5, Jacobs and Gao fail to teach inserting a client-side script into the entity body of the response as recited in claim 6.

Contrary to the Office action on page 5, Jacobs and Gao fail to describe the client-side script appends the cookie value to the URL of the web page requested to form a rewritten URL and causes the client to automatically re-request the resource with the rewritten URL as recited in claim 6. Specifically, Jacobs' col. 4, lines 10-36 and Gao paragraphs 0047-0049 fail to teach a client-side script causing a client to automatically re-request the resource with the rewritten URL.

Contrary to the Office action on pages 5-6, Jacobs col. 4, lines 10-36 fail to teach the server setting the cookie value by determining the last modified time of each web page in the same class as the web page which is the subject of the request, and setting the cookie value to the maximum value of the last modified times as recited in claims 7-8.

Contrary to the Office action on pages 6-8, Jacobs, Kavner, and Gao fail to teach claims 10, 11, and 14 for the reasons presented above with respect to claim 1 This is also true for claims 9, 12 and 13 except for the language as to a last version.

Claims 10-14 are separately patentable for the added limitations recited therein.

Contrary to the Office action on pages 8-11, Jacobs, Kavner, and Gao do not teach claims 15-22 for the reasons presented above with respect to claim 1 as well as for the additional limitations recited therein.

It is submitted that the application is in condition for allowance.

Respectfully submitted,

Robert Moll
Reg. No. 33,741

1173 Saint Charles Court
Los Altos, CA 94024
Tel: 650-567-9153
Fax: 650-567-9183
Email: rgmoll@patentplanet.com